

**BRIEF ON APPEAL**

Steven Fischman  
Attorney for Appellant  
Registration No. 34,594

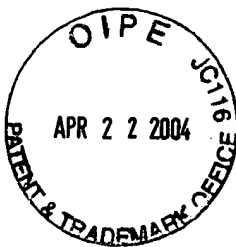
SCULLY SCOTT MURPHY & PRESSER  
400 Garden City Plaza  
Garden City, New York 11530  
(516) 742-4343

04/23/2004 MAHMED1 00000028 500510 09855240

01 FC:1402 330.00 DA

## TABLE OF CONTENTS

	<u>PAGE</u>
1. REAL PARTY IN INTEREST.....	2
2. RELATED APPEALS AND INTERFERENCES.....	2
3. STATUS OF CLAIMS.....	2
4. STATUS OF AMENDMENTS.....	2
5. SUMMARY OF THE INVENTION .....	2
6. CONCISE STATEMENT OF THE ISSUES PRESENTED FOR REVIEW .....	11
7. GROUPING OF CLAIMS.....	11
8. APPELLANT’S ARGUMENTS WITH RESPECT TO EACH OF THE ISSUES ON APPEAL.....	12
9. CONCLUSION.....	19



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

**Applicants:** Louis L. Hsu, et al.

**Examiner:** K. N. McLean Mayo

**Serial No:** 09/855,240

**Art Unit:** 2818

**Filed:** May 15, 2001

**Docket:** YOR920000660US1 (13959)

**For:** HIGH SPEED EMBEDDED DRAM  
WITH SRAM-LIKE INTERFACE

**Confirmation No.:** 9000

**Dated:** April 19, 2004

Commissioner for Patents  
P. O. Box 1450  
Alexandria, VA 22313-1450

**BRIEF ON APPEAL**

Sir:

This is a BRIEF ON APPEAL in support of applicant's appeal from the Examiner's decision dated November 19, 2003 finally rejecting claims 1, 3-10 and 12-21. This BRIEF ON APPEAL is arranged in compliance with 37 C.F.R. §1.192(c), with subheadings and the numbers of the subheadings in conformance therewith.

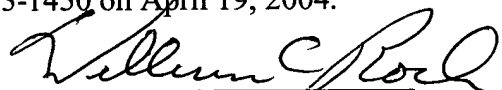
Claims 1, 3-10 and 12-21 are set forth in APPENDIX A attached hereto, as specified by 37 C.F.R. §1.192(c)(9).

---

**CERTIFICATE OF MAILING UNDER 37 C.F.R. §1.8(a)**

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner of Patents, P. O. Box 1450, Alexandria, VA 22313-1450 on April 19, 2004.

Dated: April 19, 2004

  
William C. Roch

**1. REAL PARTY IN INTEREST**

The real party in interest is the assignee International Business Machines Corporation.

**2. RELATED APPEALS AND INTERFERENCES**

No related appeals or interferences are known to the appellant, or the appellant's legal representatives, which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

**3. STATUS OF CLAIMS**

Claims 1, 3-10 and 12-21 are pending in this patent application, and are involved in this appeal. Claims 2 and 11 have been canceled.

**4. STATUS OF AMENDMENTS**

A REQUEST FOR RECONSIDERATION was filed on February 18, 2004, and was unsuccessful. No AMENDMENTS have been filed subsequent to the Final Rejection.

**5. SUMMARY OF THE INVENTION**

The present invention relates to a high speed embedded DRAM with a single port SRAM-like interface which is used in short-cycle high-speed data operations. p (page) 1, 1 (lines) 8-11

In some embedded applications, not only the speed, but also the size of the memory is critical. This is especially true for some applications, for example, a router switch, network processor, etc. where a large memory size is required. In the prior art 1T (Transistor)-SRAM, the efficiency of pipeline data flow is low, and the prior art does not

discuss sharing of internal buses to save chip area. Data congestion also appears to be a substantial problem with the design. p 1, l 19-24

The subject invention provides of a high speed embedded DRAM with a simple interface circuit between a large capacity, high speed DRAM memory and a SRAM cache to achieve a fast-cycle memory performance. The interface circuit provides wider bandwidth internal communications than external data transfers. The interface circuit schedules parallel pipeline operations so that one set of bus wiring can be shared in cycles by several data flows to save chip area and alleviate data congestion. The interface circuit utilizes a single port SRAM, instead of a dual port SRAM, which is used for short-cycle, high-speed data operations. A flexible design is provided that can be used for a range of bandwidths of data transfer. The sizes of the bandwidths indicated in the disclosed embodiment are only exemplary, and generally any size bandwidth ranging from 32 to 4096 wide can use the same approach. p 1, l 28 to p 2, l 9

Significant features of this invention can be summarized as:

- (1) providing a high-efficiency parallel-pipeline data flow so that, within each cycle, up to five tasks can be executed simultaneously,
- (2) controlling data flow in each pipeline so that a majority of the internal buses can be time shared to save chip area,
- (3) minimizing the process time of each cycle so that both latency and cycle time can be reduced, and
- (4) realizing fast-cycle, high-speed, high-density eDRAM applications without using a large sized dual port SRAM cache. p 2, l 10-18

Figure 1 is a block diagram of a high speed DRAM which includes an interface circuit designed to provide wider bandwidth data communications between a large capacity

eDRAM memory 1000 and a SRAM cache 100 than between the SRAM cache 100 and the outside world through data buses DQ. p 3, l 27 to p 4, l 2

A small single port SRAM array 100 is used as a high-speed cache between a large-sized eDRAM memory 1000 and CPU(s) (not shown) over the data buses DQ. The size of the cache 100 depends upon the architecture of the eDRAM 1000, and is generally in the range of 64K to 1M. The circuit of Figure 1 provides a wide bandwidth interface circuit between the SRAM cache 100 and the eDRAM 1000. A short distance therebetween allows a wide internal data bandwidth over wide data bus sets to improve the circuit performance. However, such wide data bus sets should be shared as much as possible. In the exemplary circuit, 512 bit (wide) bandwidth data bus sets are used between the cache 100 and the eDRAM 1000. p 3, l 3-1

Because of a restriction on the number of I/O pins, the bandwidth to the outside world is limited to 64 bits via the shared data DQ buses. p 3, l 12-13

The interface circuit couples data between the high speed DRAM 1000 and the cache memory 100 which comprises a single port SRAM. A read register 300 is coupled between the cache memory and the DRAM memory, for transferring data from the cache memory to the DRAM memory. A write register 400 is coupled between the DRAM memory and the cache memory, for transferring data from the DRAM memory to the cache memory. p 4, l 14-19

A first bi-directional data bus 1 is coupled between the cache memory 100 and both the read register 300 and the write register 400. A multiplexer 200 couples the cache memory 100 to either of the read register 300 or the write register 400. A fourth data bus 4 couples the multiplexer 200 to the read register 300, and a fifth data bus 5 couples the multiplexer 200 to the write register 400. The data flows through the bi-directional bus 1 in a first direction from the cache memory to the read register, and data flows through the bi-directional bus 1 in a second opposite direction from the write register to the cache memory, such that opposite direction data flows share the same bi-directional data bus 1 in different cycles. p 4, l 20-28

A second data bus 2 is coupled between the read register 300 and the DRAM memory 1000, and a third data bus 3 is coupled between the DRAM memory and the write register, wherein during operation data flows from the read register to the DRAM memory in one cycle, and data flows from the DRAM memory to the write register in another cycle, to share access to the DRAM memory in different cycles. p 5, l 1-5

A sixth data bus 6 couples the read register 300 to a data output from the circuit through a multiplexer 700, a ninth data bus 9, and a read buffer. A seventh data bus 7 couples a data input to the interface circuit through data lines DQ and a write buffer 500 to the write register 400. An eighth data bus 8 couples the write register 400 to a data output from the circuit, through the multiplexer 700, a read buffer 600 and the data lines DQ. A multiplexer 700 switches between inputs received from the sixth data bus 6 from the read register 300 and the eighth data bus 8 from the write register 400, and outputs data onto the ninth data bus 9 coupled to a data output from the circuit to the data lines DQ. p 5, l 6-14

A read buffer 800 couples the read register 300 to the DRAM 1000 memory through the read buffer 800 and a tenth data bus 10, and an eleventh data bus 11 couples the DRAM memory 1000 to a write buffer 900 which is coupled through the third data bus 3 to the write register 400. p 5, l 15-18 .

In the disclosed embodiment, the first, second, third, fourth, fifth, tenth, and eleventh data buses all have the same first wide data bandwidth of 512 bits, and the sixth, seventh, eighth, and ninth data buses all have the same second narrow data bandwidth of 64 bits. P 5, l 19-22

A 512 bit wide data bus is connected between the cache 100 and the read register 300 (buses 1, 4 in series) and the write register 400 (buses 1, 5 in series) via the multiplexer 200. In the following explanations, these buses are termed 512 BUS(A). The data bus 1 is bi-directional, providing for data flow both into and out of the cache 100. However, the data flows in the data bus 1 are time shared, and are always in one direction at any one time, depending upon the pipeline control. The buses 2, 3, 10 and 11 are termed 512 BUS(B). p 5, l 23-29

(For the benefit of the Board members, Webster's New World Dictionary of

Computer Terms defines a “**cache hit**” as “A successful request for data from cache memory; the data is present in the cache and does not have to be retrieved from the considerably slower main memory circuits” and defines a “**cache miss**” as “An unsuccessful request for data from cache memory; the data is not present in the cache and must be retrieved from the considerably slower main memory circuits.”)

For example, when detecting a write miss WM, as illustrated in Figure 5, or a read miss RM, as illustrated in Figure 4, old data inside the cache 100 are retired, and thus must be transferred from the cache 100 to the eDRAM 1000 via a read buffer 800. p 6, l 1-4

For a read miss RM, as illustrated in Figure 4, a new set of data are retrieved from the eDRAM 1000, not only to replace the old data in the cache 100, but also to be sent to the outside world via the output read buffer 600. Therefore, during the first cycle of data flow, data flows from the cache 100 through the 512 BUS(A) and is latched into the Read Register 300, and data coming from the eDRAM 1000 are latched into the Write Register 400 through the 512 BUS (B). In the second cycle, the directional flows of the data are reversed in the BUSES (A) and (B). p 6, l 5-11

Similarly, for a write miss WM, as illustrated in Figure 5, a new set of data are written into the cache 100 to replace the retired data, partly from the outside world (64 bit) via a write buffer 500, and the rest of the data are from the eDRAM 1000. These data are merged in the Write Register 400. Again, the bi-directional data flows time-share the buses during different cycles. p 6, l 12-16

When detecting a read hit RH, as illustrated in Figure 2, data are also transferred (nondestructively) from the cache 100 through the read register 300 to an output read buffer 600 via a MUX 700. Here, according to a column address, only a portion of the data are transferred out. p 6, l 17-20

Finally, for a write hit WH, as illustrated in Figure 3, a new set of 64 bit data are transferred to the cache 100 and overwrite the portion of the old data therein. P 6, l 21-22

Details of these operations can be understood more clearly by the following descriptions for cases including: (1) Read Hit RH, (2) Read Miss RM, (3) Write Hit WH and (4) Write Miss WM. p 6, l 23-25



Figure 2 illustrates the flow of data for a two cycle read-hit RH operation. The 512 bit data that resided in the cache 100 are read out according to the row address. These data are latched into the read register 300 based upon the column address, and only a portion (for example 64 bits) of these data are transferred out to the data DQ buses via the MUX 700 and the output read buffer 600. The whole process takes two clock cycles of the pipeline process. In the first clock cycle, data are latched in sense amplifiers of the SRAM cache 100. In the second clock cycle, data are latched and decoded in the read register 300. Details on the pipe cycles are given below. The number of cycles indicated herein is only illustrative, and alternative embodiments could use a different number of cycles. p 6, l 26 to p 7, l 6

Figure 3 illustrates the flow of data for a two cycle write hit WH operation. In this process, when the system detects the write address is in the cache 100, then it transfers 64 bit data from the data DQ buses to the cache 100. These data flow via the output write buffer 500, and are then latched into the write register 400. Note that the data only occupy a portion of the write register 400 (64 out of 512), and only this portion is written into the cache based upon column address. The rest of the data in the same row of the cache is maintained unchanged. p 7, l 7-13

The write hit WH operation, illustrated in Figure 3, takes two clock cycles to finish. In a first clock cycle, data are written into the write register 400, and then in a second clock cycle are latched into the sense amplifiers of the SRAM cache 100. p 7, l 14-17

Figure 4 illustrates the flow of data for a three cycle read miss RM operation. When the system detects that the read data is not resident in the cache 100, then immediately the old data with the same row address are written back into the eDRAM 1000. The reason is that for the fast cycle eDRAM operation, the original data are destroyed after they are read into the cache 100. This operation can be performed as described in a disclosure by Toshiaki Kiriata, et al., titled, "A Destructive Read Architecture for Dynamic Random Access Memories", as disclosed in IBM docket FIS2000-0411. Therefore, when these data are not needed in the cache, they must be written back to the eDRAM, otherwise the data will be lost. p 7, l 18-26

The write-back operation is needed for both read miss RM and write miss WM operations. As illustrated in Figure 4, while the unwanted old data are written back to the eDRAM, a new set of 512 bit data from the eDRAM with the correct row address is read into the write register 400 and then to cache 100 to replace the old data set. While retrieving these data, a portion of the data are read to the data DQ buses based upon the column address. The decoding is done in the write register 400, and from there a selected 64 bits of data are transferred to the data DQ buses via an output read buffer 600. Thus two streams of data are transferred simultaneously in two opposite paths via two sets of 512 buses (A) and (B), as shown at the left of the Figures. The read and the write registers 300, 400 are needed for the purpose of sharing these buses. For example, in the first clock cycle, the old data are latched into the cache 100 sense amplifiers, while the new data are latched in the DRAM 1000 sense amplifiers. In the second cycle, the old data are latched into the read register 300, while the new data are latched into the write register 400. At the same time, 64 bit of the data are sent to the data DQ buses and are latched into the read buffer 600. Finally, in the third cycle, the old data are written back into the eDRAM 1000, and the new data from eDRAM 1000 are transferred into the cache 100 to replace the old data. As a result, all of the 512 bit wide buses from the cache through the mux 200, register 300 and buffer 800 to the eDRAM 1000 can be time shared to save chip space. However, separate local 64 bit wide data buses may be needed to send data out to the data DQ buses. The horizontal 64 bit wide bus set group can be divided in (A), (B) and (C) bus sections, as shown at the bottom of the Figures. According to this diagram, only the (A) bus section accommodate one direction of data flow, while both the (B) and (C) bus sections accommodate bidirectional data flow and are time shared among the in and out data sets. p 7, l 27 to p 8, l 21

Figure 5 illustrates the flow of data for a three cycle write miss WM operation. When the system detects the write data address is not resident in the cache 100, then again, the old data in the same row of the cache are written back into the eDRAM 1000. In the first cycle, the old data are latched in the sense amplifiers in the cache 100, while the new data are latched in the eDRAM sense amplifiers 1000. Also, 64 bits of the new data are latched into the write register 400 via (B) and (C) bus portions of 64 bits wide. In the second cycle, the

old data are transferred to the read register 300 via the (A) bus portion of 512 bits wide. At the same time, the new data are transferred from eDRAM 1000 to the Write-Register 400 via the (B) bus portion of 512 bits wide. Inside the write register 400, based on the column address, the 512 bits of data from the eDRAM 1000 and the 64 bits data from the data DQ bus sets are merged. Finally, in the third cycle, the old data are transferred and latched into the eDRAM 1000 array, while the new data are sent to the SRAM cache 100. p 8, l 22 to p 9, l 5

**Parallel Pipeline Operation:**

The uniqueness of this arrangement is that multiple operations can proceed in a parallel manner.

Figure 6 identifies all of the pipe steps and pipe operation codes including:

Cache decode via row address (A1),

Cache signal development time is the time required to get data from a SRAM cell (B1),

Cache sense time is the time required to amplify the data and send the data out of the cache (C1),

Cache cell time is the time to write and latch data to a SRAM cell (D1),

Read Register time is the time to transfer data to the read register and park the data there (E1), DO is the time to get data from the data DQ buses from the output read buffer (F1),

DRAM decoding time is the time when receiving a row address (A2),

DRAM signal development time is the time that the bit-line receives signal from a cell (B2),

DRAM sensing time (C2),

DRAM cell time is the time to write data back to DRAM cell (D2),

Write register time is the time to send data to the write register and park the data there (E2),

the time to send data to the data DQ buses via the output write buffer (F2). p 9,

l 6-28

Therefore, a Read Hit RH operation involves A1, B1, C1, E1 and F2, a total of five pipes. A Write Hit WH involves F1, E2, A1 and D1, a total of four pipes. Here, assume that write drivers drive the data directly to the bitlines and bypass the sense amplifiers. p 10, l 1-4

For a Read Miss RM operation, three pipes proceed in parallel. The first 6-step pipe writes the old data from the cache to the DRAM, the second 6-step pipe writes the data from the DRAM to the cache, and the last single step pipe retrieves the data out. The details are described above and will not be repeated here. p 10, l 5-8

Similarly, Figure 7 shows a Write Miss WM operation. p 10, l 9

Figure 8-1 illustrates RH and WH operations proceeding simultaneously in parallel. If the memory controller can prefetch more than one command, then the RH and WH operations can be processed at the same time. Otherwise, a pipe delay is required. p 10, l 10-13

Figure 8-2 illustrates WH and RM operations proceeding simultaneously in parallel. p 10, l 14-15

Figure 8-3 illustrates that two pipe delays are required for the RH and WM operations, and vice versa. p 10, l 16-17

Figure 8-4 also shows that two pipe delays are required for RM and WM operations, and vice versa. p 10, l 18-19

These are the four combinations that could happen for any two consecutive operations. Based on this, the pipe delay can be easily estimated for the other 12 possible combinations. p 10, l 20-22

Figure 9 is a summary of the pipe delays for 16 possible combinations of operations. p 10, l 23-24

One purpose of defining such a fine pipe stage is to provide high-efficiency parallel processing. As shown in Fig. 8-4, for example, the maximum number of operations of the parallel process is five. The worst case latency and consequent delay will be five and two, respectively. Since each stage is short, with today's technology, 2ns per stage is a

reasonable estimation. Therefore, this design can achieve 10ns latency and 2ns (0 pipe) to 4ns (2 pipe) data cycle time. p 10, l 25 to p 11, l 2

Further improvements are also possible based upon the same concepts, including a multiple instruction process, a dual clock rate, I/O data interleaving, etc. p 11, l 3-4

Significant features of this invention can be summarized as:

(1) providing a high-efficiency parallel-pipeline data flow so that, within each cycle, up to five tasks can be executed simultaneously,

(2) controlling data flow in each pipeline so that a majority of the internal buses can be time shared to save chip area,

(3) minimizing the process time of each cycle so that both latency and cycle time can be reduced, and

(4) realizing fast-cycle, high-speed, high-density eDRAM applications without using a large sized dual port SRAM cache. p 11, l 5-13

## **6. CONCISE STATEMENT OF THE ISSUES PRESENTED FOR REVIEW**

Whether claims 1, 3-10 and 12-21 are unpatentable under 35 U.S.C. §103 as being obvious over Leung (U.S. 6,415,353).

## **7. GROUPING OF CLAIMS**

Claims 1 sets forth a basic independent apparatus claim describing the present invention. However, each of dependent method of operating claims 13-21 specify different operations of the high speed DRAM of claim 1, and are believed to be present separate issues of patentability with respect to the prior art, and the patentability thereof are argued separately in Section 8 of this BRIEF ON APPEAL.

**8. APPELLANT'S ARGUMENTS WITH RESPECT TO EACH OF THE ISSUES**  
**ON APPEAL**

Independent claim 1 reads upon the disclosed embodiment of Figure 1 as follows.

A high speed DRAM, comprising:

a DRAM memory (1000);

a cache memory comprising a single port SRAM (100);

a read register (300) coupled between the cache memory and the DRAM memory, for transferring data from the cache memory to the DRAM memory;

a write register (400) coupled between the DRAM memory and the cache memory, for transferring data from the DRAM memory to the cache memory;

a first bi-directional data bus set (1, 4, 5) coupled between the cache memory and both the read register and the write register, wherein data flows through the bi-directional bus in a first direction from the cache memory to the read register, and data flows through the bi-directional bus in a second opposite direction from the write register to the cache memory, such that opposite direction data flows share the same bi-directional data bus in different cycles;

a second data bus set (2, 10) coupled between the read register and the DRAM memory;

a third data bus set (3, 11) coupled between the DRAM memory and the write register.

### Leung Compared To The Present Invention

The design objective of Leung is basically to hide a data refresh cycle of a DRAM, whereas the design objective of the present invention is to minimize the cycle time to access data of a DRAM/SRAM cache over very wide data buses, which are entirely different design objectives that result in entirely different architectures.

A key difference between the cache interface architecture design (more specially about the data path design) of the present invention and other prior art designs is that the design objective of the present invention is to minimize the number of clock cycles for either read or write operations, including both hit and miss situations. This has not even been discussed in any of the prior art.

In order to fulfill this design objective, the present invention uses (1) the bi-directional 512 data path at the neck region and a MUX 200, (2) a write buffer 500 directly connected between DQ and Write register 400, (3) a MUX 700 taking inputs either from Read Register 300 or Write register 400 to a read buffer 600 to send them to DQ.

The architecture of the present invention, with the very wide data buses, uses pipe-line operations. Referring to Figure 1 of this application, the very wide bi-directional data bus 1, connecting the SRAM cache 100 through MUX 200 to either read register 300 or write register 400, will not support simultaneous read and write operations.

The described architecture of Leung supports simultaneous read and write operations, but Leung does not have a similar bi-directional data bus, and instead uses two separate unidirectional data buses, dedicated unidirectional read data bus DB[255:0] and dedicated unidirectional write data bus DA[255:0].

This patent application has a single independent claim 1, with claims 2-21 being dependent upon independent claim 1. Independent claim 1 specifies in lines 8-9, “a first bi-directional data bus set coupled between the cache memory and both the read register and the write register”. This bi-directional data bus is shown as bus 1, which communicates through MUX 200, with either Read Register 300 or Write Register 400, and is a significant component of the present invention for communication between high speed DRAM 1000 and the SRAM cache 200 in the very simple arrangement of Figure 1.

Leung discloses two separate embodiments in Figures 1 and 5.

The embodiment of Figure 1 appears to use a single port SRAM in view of the statement in col. 8, lines 56-59, “In another embodiment, SRAM cache 187 is fabricated using dual-port SRAM cells, which can be used to support read and write operations during a single cycle of the CLK signal.”

However, as explained in col. 8, lines 53-56, “Cache read buffer 188 and cache write buffer 189 enable SRAM cache 187 to perform a read operation and a write operation during the same cycle of the CLK signal.”



The embodiment of Figure 5 uses a dual port SRAM/DRAM having a read-write port 313 and a write only port 312. This embodiment is also designed to perform simultaneous read and write operations during the same clock cycle.

The single port SRAM embodiment of Figure 1 of Leung appears to be more pertinent to the single port SRAM of the present invention, and accordingly only the embodiment of Figure 1 is analyzed herein as the more pertinent embodiment.

It should be recalled that the major design objective of Leung is to handle “refresh operations in a semiconductor memory such that the refresh operations do not interface with external access operations.” Col. 1, lines 32-35

Accordingly, the data bus structures of Leung are designed to accommodate that major object.

As stated above, in the embodiment of Figure 1, the cache read buffer 188 and cache write buffer 187 enable SRAM cache 187 to perform simultaneous read and write operations during the same clock cycle. The performance of simultaneous read and write operations during the same clock cycle, through the unidirectional dedicated Read bus DB[255:00] connected between the read buffer and data latches 171 of the DRAM 1000 and the write buffer 189 of the SRAM cache 187, and through the unidirectional dedicated Write

DA[255:00] connected between the read buffer 188 of the SRAM cache 187 and the write buffer and data latches 172 of the DRAM 1000.

The 256 bit wide, unidirectional dedicated Read and Write buses DB[255:00] and DA[255:00] were chosen to allow simultaneous read and write operations during the same clock cycle.

The Final Rejection attempts to read claim 1 on Leung as follows.

1. A high speed DRAM, comprising:
  - a DRAM memory (1000);
  - a cache memory comprising a single port SRAM (187);
  - a read register (188) coupled between the cache memory and the DRAM memory, for transferring data from the cache memory to the DRAM memory;
  - a write register (189) coupled between the DRAM memory and the cache memory, for transferring data from the DRAM memory to the cache memory;
  - a first bi-directional data bus set (187 to 188, 189 to 187) coupled between the cache memory (187) and both the read register (188) and the write register(189), wherein data flows through the bi-directional bus in a first direction from the cache memory to the read register, and data flows through the bi-directional bus in a second opposite direction from the write register to the cache memory, such that opposite direction data flows share the same bi-directional data bus in different cycles;

a second data bus set (DA[255:0]) coupled between the read register and the DRAM memory;

a third data bus set (DB[255:0] and bus from 193 to 189) coupled between the DRAM memory and the write register.

The Final Rejection suggests that it would be obvious in Leung to combine the bus from SRAM cache 187 to read buffer 188 with the bus from write buffer 189 to the SRAM cache 187 as one common bi-directional bus that is shared between opposite direction transfers of data to the read buffer 188 and from the write buffer 189.

This position completely ignores the reality that the data transferred from the SRAM cache 187 to the read buffer 188 is also transferred over the unidirectional dedicated Write bus DA[255:0], and the data transferred to the SRAM cache 187 from the write buffer 189 is also transferred over the unidirectional dedicated Write bus DB[255:0].

The data buses DA[255:0] and DB[255:0] were selected as unidirectional dedicated buses to allow the major object of Leung which is to handle refresh operations in a semiconductor memory such that the refresh operations do not interfere with external access operations.

The modification suggested in the Final Rejection would be incompatible with the existing buses DA[255:0] and DB[255:0].

More importantly, the modification suggested in the Final Rejection would thwart and be incompatible with the major object of Leung which is to handle refresh operations in a semiconductor memory such that the refresh operations do not interfere with external access operations.

Claims 13-20 define methods of operating the high speed DRAM of claim 1 for respective operations of a read miss (13), write miss (14), read hit (15), write hit (16), two cycle read hit (17), two cycle write hit (18), three cycle read miss (19), and three cycle write miss (20).

Leung is simply not concerned with high efficiency parallel-pipeline data flow operations, and so does not disclose or teach the subject matter of claims 13-20

Figure 5 explains clearly how data flow in a write miss case. No known prior art data interface will facilitate such a write miss data transfer in a pipe-line fashion.

A write miss operation is defined as writing a word line of data into the cache 100 which is 512 bits wide from the external DQ. But only 1/4 of the word line of data needs to be written, and the other 3/4 is taken from the eDRAM. At this point, the 512 bits from eDRAM is loaded to write buffer 900 and 64 bits of new data is loaded to write buffer 500 and both of them are mapped to write register 400, the 64 bits of the new data will overwrite

the 64 bits of the old 512 bit data and the whole modified 512 bits data will be send to cache 100 through MUX 200. At the same time, the old 512 bit data from the cache 100 are retired from the cache 100 back to the eDRAM 1000.

This type of write miss operation is entirely novel relative to the prior art and Leung.

## 9. CONCLUSION

In view of the above, it is respectfully submitted that the Final Rejection is in error and should be reversed for good reasons, and it is respectfully requested that the Board of Patent Appeals and Interferences so find.

Respectfully submitted,

A handwritten signature in black ink, appearing to read 'Steven Fischman', with a long horizontal flourish extending to the right.

Steven Fischman  
Registration No. 34,594

SCULLY, SCOTT, MURPHY & PRESSER  
400 Garden City Plaza  
Garden City, New York 11530  
(516) 742-4343

WCR/jf